

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Ilija Marinović

Zagreb, 2017.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Bojan Jerbić, dipl. ing.

Student:

Ilija Marinović

Zagreb, 2017.

Izjavljujem da sam ovaj rad napisao samostalno koristeći stečena znanja na fakultetu kao i stručnu literaturu.

Zahvaljujem se prof. dr. sc. Bojanu Jerbiću što mi je bio mentor i što mi je omogućio izradu završnog rada u području robotike te na velikom strpljenju, uloženom trudu i pomoći pri izradi završnog rada. Zahvaljujem se i Filipu Šuligoju mag.ing.mech na konstruktivnim kritikama i pomoći pri izradi završnog rada. Zahvaljujem se svim djelatnicima u zavodu za robotiku i automatizaciju proizvodnih sustava na pomoći prilikom izrade završnog rada te firmi Klimaoprema d.d. koja je omogućila određenu opremu i razvila interes za ovo područje strojarstva.

Zahvaljujem se roditeljima, obitelji i prijateljima na podršci tijekom studija .

Ilija Marinović



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za završne ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo
materijala i mehatronika i robotika

Sveučilište u Zagrebu	
Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

ZAVRŠNI ZADATAK

Student:

ILIJA MARINović

Mat. br.: 0035193433

Naslov rada na
hrvatskom jeziku:

**Automatski sustav za određivanje parametara vizijske kontrole grešaka
na limu**

Naslov rada na
engleskom jeziku:

**Automatic system for detection of vision system parameters in sheet
metal inspection**

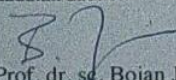
Opis zadatka:

Kod primjene vizijskog sustava za kontrolu limova potrebno je precizno odrediti komponente sustava i njihove parametre kako bi se sve vrste grešaka mogle uočiti na različitim tipovima limova. Za prethodno odabrane komponente vizijskog sustava (Baslerova linescan kamera, lotus linelight, Hengstler davač položaja, itd.) potrebno je odrediti optimalne fizičke parametre takvog tehničkog sustava. Fizički parametri sustava su udaljenosti kamere i rasvjetnog sustava od lima, kut kamere i svjetla u odnosu na lim i intenzitet svjetla. Za određivanje najpovoljnijih fizičkih parametara vizijskog sustava potrebno je izvesti automatski sustav koji će samostalno izvoditi veliki broj testiranja za različite konfiguracije fizičkih parametara. Zadatak je povezati i programirati automatski sustav koji se sastoji od 2 robota (1 nosi kameru, a drugi svjetlo), pokretne trake, računala (koristi se postojeći c++ program za primanje, procesiranje i pohranu slike), kamere, svjetla i davača položaja. Parametri sustava kao i rezultati vizijske inspekcije moraju se automatski pohranjivati na računalu.

Zadatak zadan:

9. studenog 2016.

Zadatak zadao:


Prof. dr. sc. Bojan Jerbić

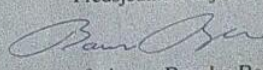
Rok predaje rada:

1. rok: 24. veljače 2017.
2. rok (izvanredni): 28. lipnja 2017.
3. rok: 22. rujna 2017.

Predviđeni datumi obrane:

1. rok: 27.2. - 03.03. 2017.
2. rok (izvanredni): 30. 06. 2017.
3. rok: 25.9. - 29. 09. 2017.

Predsjednik Povjerenstva:


Izv. prof. dr. sc. Branko Bauer

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	II
POPIS TABLICA.....	III
SAŽETAK.....	IV
SUMMARY	V
1. UVOD.....	1
2. ELEMENTI SUSTAVA.....	2
2.1 KAMERA.....	3
2.2 SVJETLO I OSVJETLJENJE.....	5
2.3 ENKODER.....	7
2.4 ROBOTI.....	8
2.4.1 Universal robots UR3.....	10
2.4.2 Universal robots UR5.....	14
3. TEHNIČKE SPECIFIKACIJE SUSTAVA.....	17
3.1 POSTAVKE SLIKE I KAMERE	17
3.2 KOMUNIKACIJA IZMEĐU ROBOTA	20
3.3 KOREKCIJA POMAKA USLJED ROTACIJE SVJETLA	23
4. ALGORITAM I PROGRAMI.....	26
4.1 ALGORITAM AUTOMATSKOG PROGRAMA	29
4.2 PROGRAM ZA AUTOMATSKO SNIMANJE NA ROBOTU UR5	30
4.3 PROGRAM ZA AUTOMATSKO SNIMANJE NA ROBOTU UR3	32
4.3 UR5 PROGRAM ZA PRECIZNIJE POZICIONIRANJE.....	33
4.4 UR3 PROGRAM ZA PRECIZNIJE SNIMANJE	36
5. REZULTATI SNIMANJA	38
6. ZAKLJUČAK.....	42

POPIS SLIKA

Slika 1: Shema tehničkog sustava	2
Slika 2: Area kamera i line kamera	3
Slika 3: Kamera raL4096-24gm	4
Slika 4: Usporedba vise izvora svjetla	5
Slika 5: Tipovi osvjetljenja	6
Slika 6: Ilustracija problema i položaja komponenata	6
Slika 7: Prikaz svjetla u laboratoriju	7
Slika 8: Enkoder RI58-O/10000EK.42KB-F0 u našem sustavu	8
Slika 9: Predviđeni rast broja robota u svijetu [5]	9
Slika 10: Robot UR3 s montiranom kamerom	9
Slika 11: Universal robot UR3	10
Slika 12: Tehničke specifikacije UR3 robota [6]	11
Slika 13: Spajanje tipkala na kontroler robota	11
Slika 14: Kontroler robota s izvučenim tipkalima za precizno snimanje	12
Slika 15: Središnja točka alata (kamere)	13
Slika 16: Mrežne postavke UR3 robota	13
Slika 17: Robot UR5 s improviziranom hvataljkom za svjetlo	14
Slika 18: Središnja točka alata UR5	15
Slika 19: Mrežne postavke robota UR5	16
Slika 20: Shema toka signala(isprekidana) i informacija(puna)	17
Slika 21: Smjer snimanja slike	18
Slika 22: Signali s kamere	18
Slika 23: Primjer slike nakon postavljanja parametra	19
Slika 24: Modbus klijent	20
Slika 25: Postavljanje modbus komunikacije	21
Slika 26: Adrese portova i njihova namjena	21
Slika 27: Odabrane adrese registara	22
Slika 28: Pogreške kod snimanja zbog crne podloge	23
Slika 29: Neispravno i ispravno podešeno snimanje	24
Slika 30: Letva s vijcima	25
Slika 31: Otvaranje liste programskih instrukcija	26
Slika 32: Prikaz nekih instrukcija	27
Slika 33: Oprecije naredbe <i>Wait</i>	28
Slika 34: Opcije naredbe <i>Set</i>	28
Slika 35: Kutovi s najboljim rezultatima	38
Slika 36: Primjer snimanja 1	39
Slika 37: Primjer snimanja 2	39
Slika 38: Primjer snimanja 3	40
Slika 39: Finalna fotografija	40
Slika 40: Analiza snimljene fotografije	41

POPIS TABLICA

Tablica 1: Specifikacije kamere [1]	4
Tablica 2 : Prikaz komunikacije između robota	22

SAŽETAK

U ovom radu obrađen je automatski tehnički sustav za određivanje optimalnih parametara vizijske inspekcije lima. S obzirom da je kontrola kvalitete lima nužna kako bi se u ranoj fazi proizvodnje pronašle greške i smanjili budući gubici, potrebno je pronaći optimalne parametre snimanja da bi greške bile što uočljivije. Tražene greške su površinska oštećenja u vidu točaka ili linija te udubljenje lima. Kako sve greške nisu jednako vidljive pod svim kutovima svjetla i kamere, traži se njihov optimalni raspon. Pronalazak toga raspona u ručnom načinu rada je mukotrpan posao jer je potrebno osigurati paralelnost kamere, svjetla i lima uz promjenu kuta, iz tog razloga dolazi potreba za automatizacijom ovog procesa. Zadatak rada bio je napraviti tehnički sustav koji bi integrirao dva robota, transportnu traku, računalo, kameru i enkoder. Automatizacija procesa uključuje automatsku izmjenu različitih tehničkih parametara te snimanje i evaluaciju. Napravljena su dva režima rada, prvi, za automatsku pretragu područja u kojem su greške najbolje vidljive i drugi koji služi sa preciznije postavljanje parametara.

U prvom dijelu završnog rada predstavljani su elementi ovog tehničkog sustava te je prikazana njegova shema. Drugi dio se odnosi na komunikaciju između pojedinih elemenata, također prikazati će se neka rješenja za nailazeće probleme. U završnom dijelu bit će prikazani robotski programi te rezultati snimanja.

Ključne riječi: robot, automatizacija, vizijska inspekcija, lim, kamera.

SUMMARY

This paper deals with an automatic technical system for determining the optimal parameters of the visual inspection of the sheet metal. Since quality control is necessary to find errors at an early stage of production and to reduce future losses, it is necessary to find the optimal recording parameters to make the errors as noticeable as possible. The required errors are surface damage in the form of dots or lines, and sheet metal recesses. Since all the faults are not equally visible at all light angles and camera, their optimal range is required. Finding this range in manual mode is a hard-labouring job due to the fact that it is necessary to ensure the parallelism of the camera, the light, and the file with the angle change, therefore it is necessary to automate this process. The aim of this work was to create a technical system that would integrate two robots, a carrying strap, a computer, a camera and an encoder. Automation of the process involves automatical modification of various technical parameters, recording and evaluation. Two modes of operation were created, the first to automatically search the area where the errors are most visible and the other one that serves the most accurate parameter setting.

In the first part of the final paper are presented the elements of this technical system and its scheme. The second part refers to communication between individual elements and some problem solutions.. The final part will show the robot programs and the results of the recordings.

Keywords: robot, automation, vision inspection, sheet metal, camera.

1. UVOD

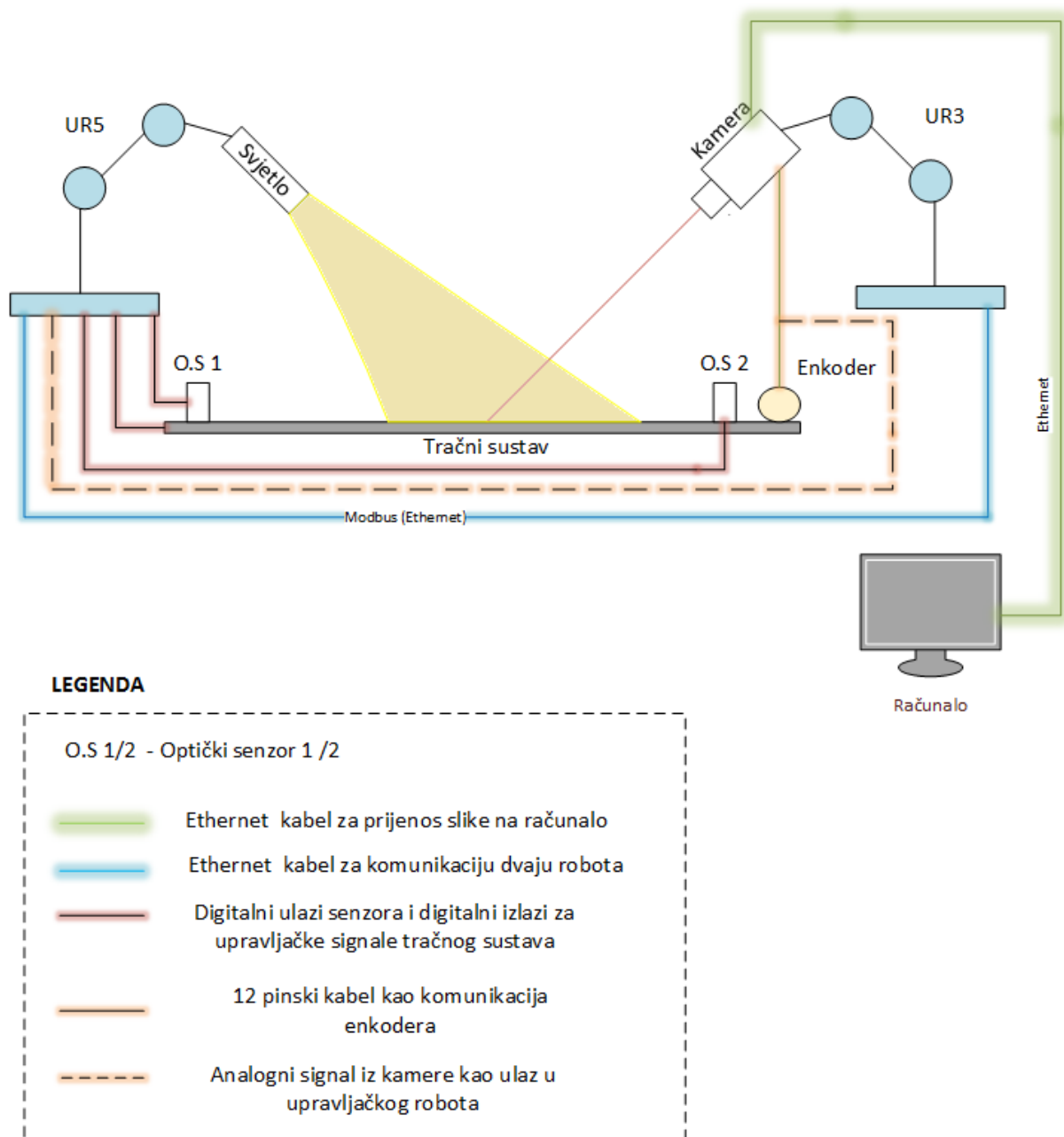
U moderno doba teško je govoriti o visokokvalitetnoj industriji i proizvodima, a da ne spomenemo pojam automatizacije. Tržišno nadmetanje ljestvicu kvalitete i proizvodnih količina diže na više razine te dolazimo u područje ljudskih limita. Iz tog razloga industrije se, gdje god je to moguće i ekonomski opravdano, okreću automatizaciji tj. robotizaciji proizvodnje.

Trend relativno niskih cijena manipulatora i robota učinio je visoku tehnologiju pristupačnom srednjim i malim tvrtkama. Uvođenjem takvih sustava povećava se proizvodna količina, ušteda materijala te se postiže ujednačena kvaliteta proizvoda. Upravo je kvaliteta segment na koji treba obratiti posebnu pozornost jer što se greške kasnije uoče veći su gubitci materijala i vremena, a može doći i do zastoja proizvodne linije, stoga se veliki naponi ulažu u kontrolu kvalitete proizvoda, a tu je veliki pomak odigrala i pojava strojnog odnosno računalnog vida. Vizijski sustavi zauzimaju veliki udio u kontroli u prehrambenoj industriji gdje se provjeravaju pakiranja i njihov sastav, montaži gdje je potrenno precizno pozicioniranje pojedinih dijelova kao i kontroli kvalitete određenih površina, a upravo taj dio odnosno pronalazak grešaka na limu okupacija je ovog rada. Vizijski sustavi koji kombiniraju opremu za snimanje s algoritmima za pronalaženje grešaka postaju sve bolji, no uz sve napore sustavi detektiraju tek oko 50% - 65% grešaka. Kao razlog tako niske djelotvornosti procesa često se navode šumovi uslijed neadekvatnog osvjetljenja pa se veliki naponi ulažu u unaprijedjenje tih sustava, a mali doprinos tome dati će i ovaj rad.

Ako se promatra čovjek kao reprezentativni organizam uviđa se kako su oči organ kojim primamo najviše informacija. Stoga ako se taj organ donira robotu odnosno računalu u obliku kamere otvaramo mu mogućnost rada u operacijama koje su do tada radili isključivo ljudi, međutim, uspješnost kvalitetne vizijskih sustava je u odabranim parametrima kao što su intenzivnost i položaj svjetla te udaljenost svjetla i kamere od predmeta. Malo odstupanje od optimalnog daje potpuno drugačije rezultate, a pronalazak optimalnih parametara iziskuje puno napora, pa je zadatak ovog rada bio napraviti automatski sustav za pronalaženje optimalnih parametara položaja kamere i svjetla kod vizijske inspekcije lima.. Kao rješenje na postavljeni zadatak sastavljen je i konfiguriran tehnički sustav koji su kombinaciji s napisanim robotskim kodom te unaprijed razrađenim kodom za vizijsku inspekciju lima automatski pronalazi optimalne parametre snimanja.

2. ELEMENTI SUSTAVA

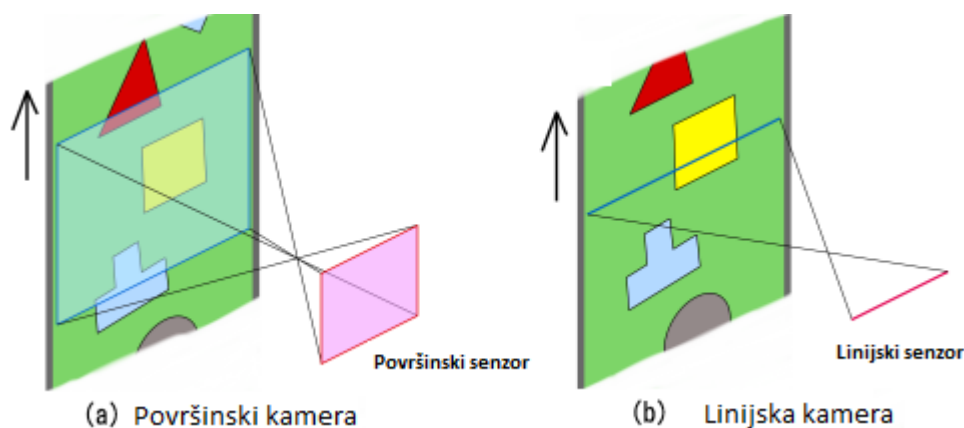
U samom uvodu ovaj rad se predstavlja kao tehnički sustav koji integrira dva robota, pokretnu traku, računalu, kameru, optičke senzore i enkoder. Iz tog razloga slika 1 prikazuje shemu sustava kako bi se jasnije očitala uloga pojedine komponente u sustavu.



Slika 1: Shema tehničkog sustava

2.1 KAMERA

Kamera je, uz svjetlo, najbitniji element našeg sustava te o njoj ovisi kvaliteta slike koju će računalo procesirati. U industriji se za ovakve probleme koriste dva tipa kamera, kamera s površinskim senzorom i linijska kamera. Razlika je ta što kamera s površinskim senzorom snima površinski segment. Da bi se s takvom kamerom snimao predmet u pokretu, kamera treba snimati preklapajuće slike koje bi kasnije trebalo softverski rezati da bi se postigao pravi redoslijed slike.



Slika 2: Area kamera i line kamera

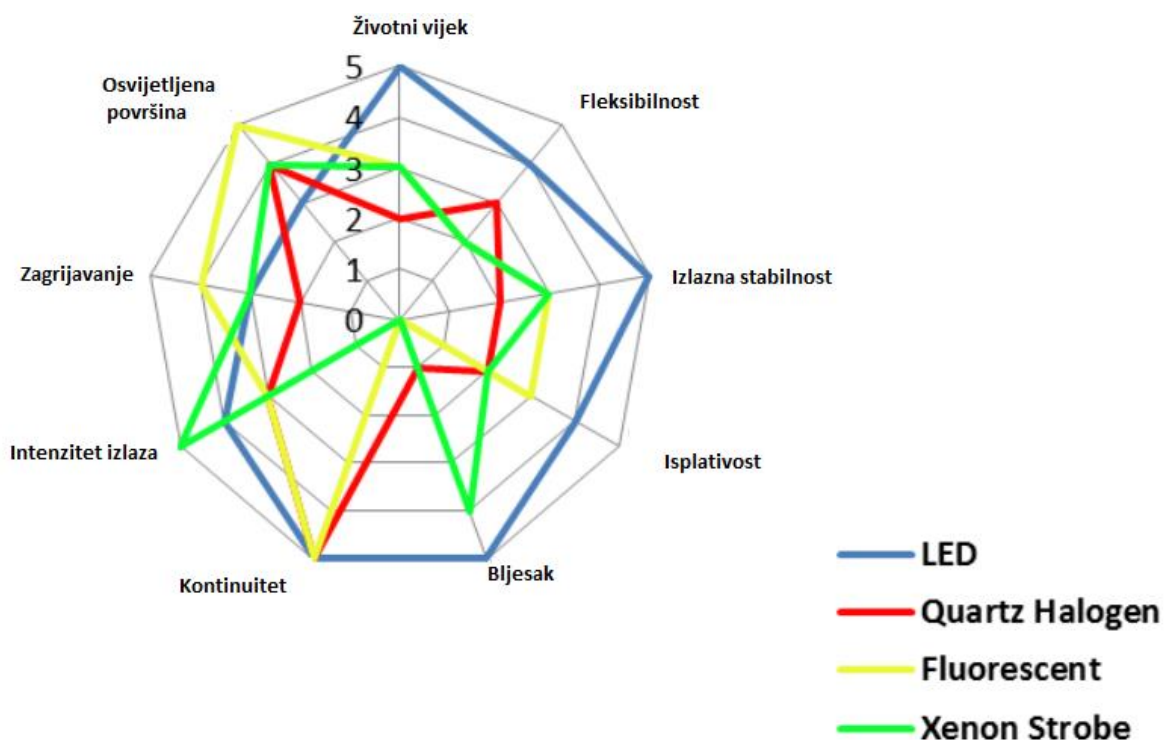
Linijska kamera, s druge strane, ima samo jedan red fotoosjetljivih piksela koji kontinuirano snimaju objekte što se kreću po konvejeru. U praksi, vrijednost pojedinog piksela se konvertira u digitalne vrijednosti te se učitava u vodoravni registar. To može biti pohranjeno i obavljeno od strane računala, stoga je za potrebe ovoga rada bila odabrana linijska kamera proizvođača Basler, a model kamere je raL4096-24gm čije su specifikacije u tablici 1.

**Slika 3: Kamera raL4096-24gm****Tablica 1: Specifikacije kamere [1]**

Sepcifikacije	raL4096-24gm
Veličina senzora	4096 piksela
Tip senzora	Awaiba DR-4k-7
Veličina piksela	7 μm x 7 μm
Max line rate	24 KHz
I/O	3 ulaza i 2 izlaza
Sinkronizacija	putem vanjskog okidnog sigala
	putem softvera
	slobodan rad

2.2 SVJETLO I OSVJETLJENJE

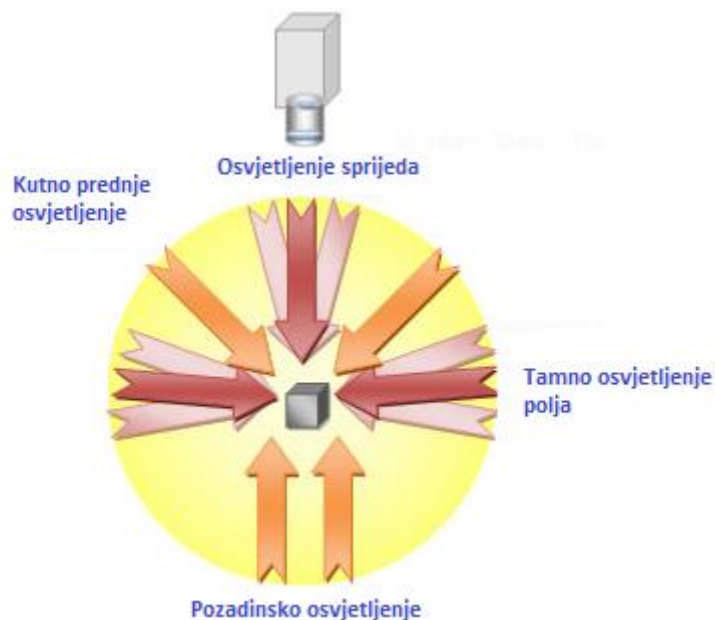
Kao što je prethodno bilo spomenuto, svjetlo je jedan od izvšnih elemenata ovog sustava. Cilj ovog rada je bio automatizirati sustav za određivanje optimalnih parametara za dobivanje što kvalitetnije slike. Kvaliteta slike uvelike ovisi o tipu svjetla kao i o osvjetljenju pa je potrebno odabrati ispravan tip svjetla i osvjetljenja. Kako je navedeno u [2] i [3] LED svjetla imaju niz prednosti u odnosu na druge izvore svijetlosti, međutim imaju i neke nedostatke poput manje posvijetljene površine, slika 4.



Slika 4: Usporedba više izvora svjetla

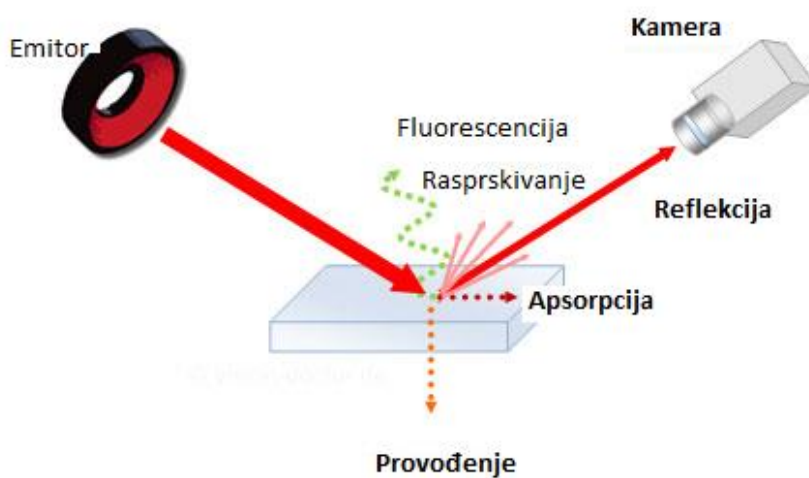
Kako je ranije spomenuto da se koristi linijska kamera veličina, osvjetljenje površine nije najbitniji parametar. S druge strane, potrebno je paziti i na način osvjetljenja predmeta, a u ovom slučaju to je komad lima različitih boja. Upravo ta različnost boja može uzrokovati probleme kod osvjetljenja jer će bijela podloga više svjetla reflektirati od npr. zelene. Također, u industrijskoj primjeni lim je prevučen tankom folijom koja stvara dodatne probleme kod refleksije.

Neke od tehnika osvjetljenja koje se koriste u praksi su: pozadinsko osvjetljenje, kupolno osvjetljenje, direktno osvjetljenje, osvjetljenje pod kutom te osvjetljenje pod malim kutom (tzv. osvjetljenje tamnih površina).



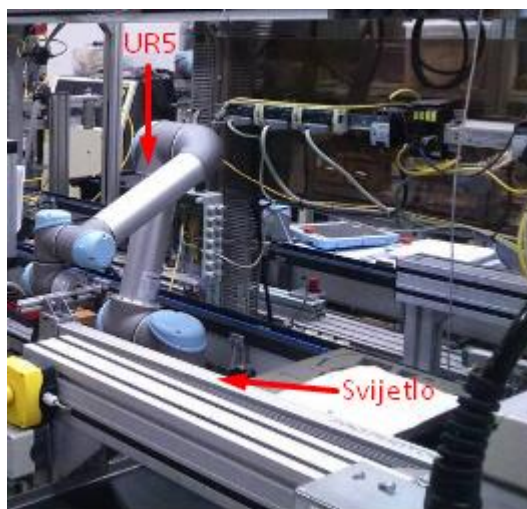
Slika 5: Tipovi osvjetljenja

Zbog konfiguracije sustava promatrati ćemo zadnja dva tipa osvjetljenja tako da će robot koji drži kameru biti na jednom kraju pokretne trake, a svjetlo kao što je to ilustrativno prikazano na slici 6.



Slika 6: Ilustracija problema i položaja komponenata

Svjetlo koje se koristilo u ovome radu je Prophononix LOTUS intenzivno bijelo linijsko svjetlo. Svjetlo ima pasivno hlađenje i mase je 13 kg, a s obzirom da je za nošenje svjetla na raspolaganju samo robot UR5, morat će se jedan kraj svjetla uležištiti na nosivu konstrukciju u laboratoriju dok će drugi kraj nositi robot te rotirati pomoću rotacije oko z osi.



Slika 7: Prikaz svjetla u laboratoriju

2.3 ENKODER

Enkoderi su uređaji koji služe kao mjerni pretvarači za pretvaranje kutnog ili linearnog pomaka. Postoje dvije vrste enkodera: apsolutni i inkrementalni, apsolutni enkoderi pamte poziciju tj. na izlazu daju binarni kod pozicije u kojoj se nalaze. Kod inkrementalnih enkodera rezoluciju određuje broj impulsa po jednom okretaju osovine. U ovome radu koristio se Hengstler inkrementalni enkoder RI58-O/10000EK.42KB-F0 rezolucije 10 000 impulsa [4] prikazan na slikci 8. Enkoder je montiran na jedan kraj transportne trake pomoću nosača i inprovizirane prirubnice, također spojen je 12 pinskim kabelom i na kameru pomoću koje on zapravo okida sliku nakon određenog broja okretaja. Isto tako jedan analogni izlaz je spojen u robota kako bi se dao znak da je slikanje završeno, u tom slučaju robot gasi te mijenja smjer kretanja transportne trake.

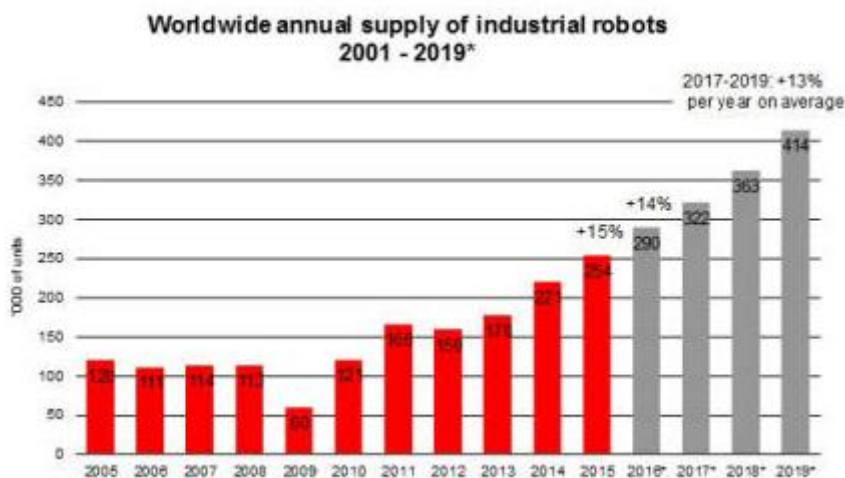


Slika 8: Enkoder RI58-O/10000EK.42KB-F0 u našem sustavu

2.4 ROBOTI

Roboti su upravljivi reprogramibilni višenamjenski manipulatori s tri ili više osi. Robot može biti stacionaran ili pokretan. Mehanička struktura robota se sastoji od više povezanih zglobova, upravo su zglobovi ti koji robotu daju mobilnost (stupnjeve slobode). U zglobovima se nalaze motori i enkoderi koji pamete položaje pojedinog zgloba.

Roboti su bitan čimbenik u automatiziranoj industriji posebno u pogledu montaže gdje se zapravo gubi najviše vremena za relativno monoton proces. Budući da se broj robota u svijetu povećava iz godine u godinu, nije teško zaključiti da je bez robotizirane proizvodnje teško pratiti tempo konkurencije.



Slika 9: Predviđeni rast broja robota u svijetu [5]

S obzirom da se naš sustav sastoji od dva izvršna člana (kamera i svjetla), oni moraju biti prostorno raspoređeni tako da se zadovolji promjenjiv kut upada svjetla i kamere. Takvu vrstu fleksibilnosti najlakše je postići korištenjem robota koji će nositi te komponente. Kako su na raspolaganju dva robota, UR5, nosivosti 5kg i UR3, nosivosti 3 kg. S obzirom da je masa kamere relativno mala, ona će se montirati na UR3 dok će robot UR5 nositi svjetlo veće mase od svoje nosivosti na način koji je objašnjen u poglavlju 2.2.



Slika 10: Robot UR3 s montiranom kamerom

Transportna traka može biti upravljana PLC-om, ali se u ovom radu išlo na opciju da traka bude upravljana pomoću jednog od robota kako bi se imala bolja kontrola nad programom te kako bi bilo lakše ostvariti komunikaciju između dvaju robota u procesu. Roboti međusobno komuniciraju i povezani su ethernet kabelom, dok im je konfiguracija napravljena putem modbusa što će detaljno biti objašnjeno kasnije.

2.4.1 Universal robots UR3



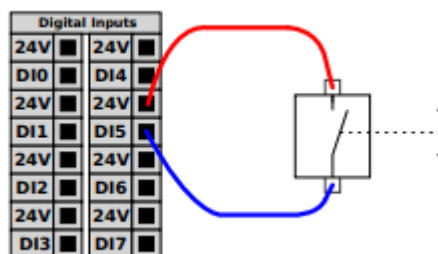
Slika 11: Universal robot UR3

UR3 je najmanja verzija industrijskih robota kompanije Universal robots sa sjedištem u Danskoj. Robot je nosivosti 3 kg i radijusa rada 0,5 m. Odobren je TUV certifikatom koji jamči sigurnost i kvaliteu, te može raditi u interakciji s čovjekom jer posjeduje senzore sile koji zaustavlja robota čim naiđe na prepreku. Universal robot je jednostavno programirati jer dolaze s tabletom osjetivim na dodir na kojem se nalaze sve instrukcije te 3D vizualizacija robota. Također, postoji i softver pomoću kojega je moguće offline programiranje na računalu, software je potpuno besplatan i može ga se pokrenuti pomoću virtualnog operacijskog sustava.

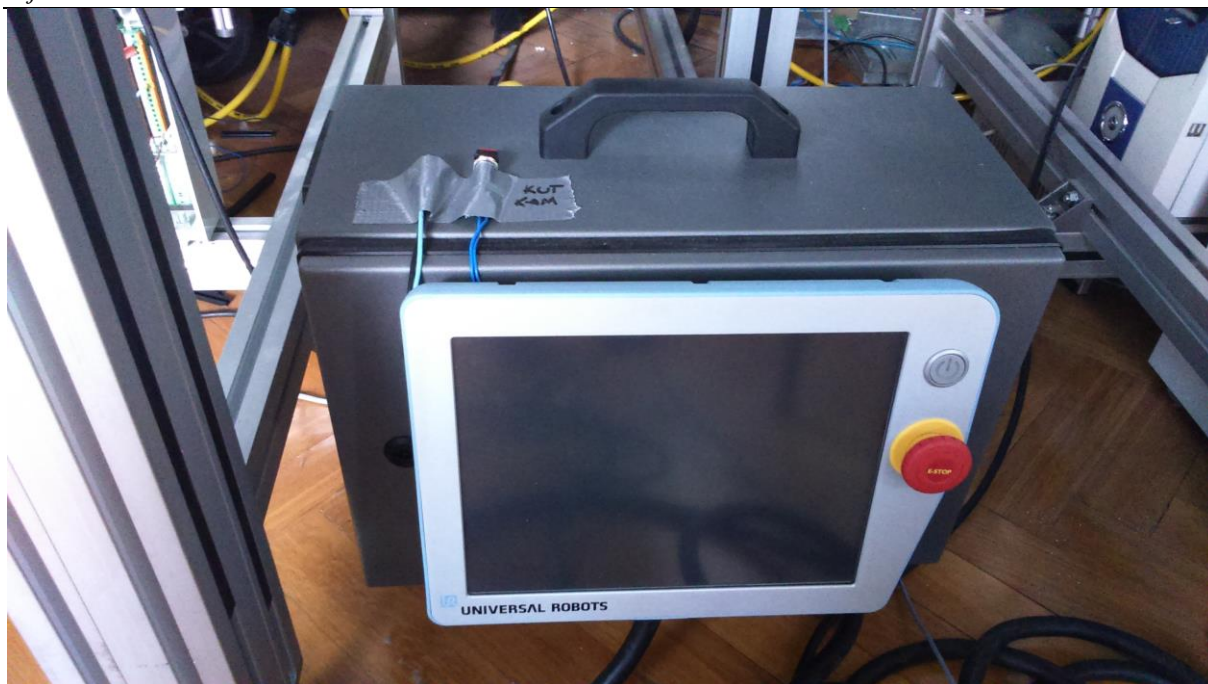
UR3	
Performanse	
Ponovljivost	±0.1 mm / ±0.0039 in
Radna temperatura	0-50* *
Potrošnja	Min 90W, Typical 125W, Max 250W
Specifikacije	
Nosivost	3 kg / 6.6 lbs
Radni doseg	500 mm / 19.7 in
Stupnjevi slobode	6 stupnjeva slobode gibanja
Programiranje	Polyscope grafičko sučelje na privjesku za učenje

Slika 12: Tehničke specifikacije UR3 robota [6]

U kontroleru robota postoji osam digitalnih te dva analogna ulaza i izlaza te još osam programibilnih ulaza i izlaza. Kako je slobodnom voljom odlučeno da će UR5 biti robot koji će upravljati sustavom na ovaj kontroler, spojena su samo tipkala koja služe za precizno namještanje kamere u pomaku od 5°. Tipkala su spojena na digitalne ulaze kao što je to prikazano na slici 13.

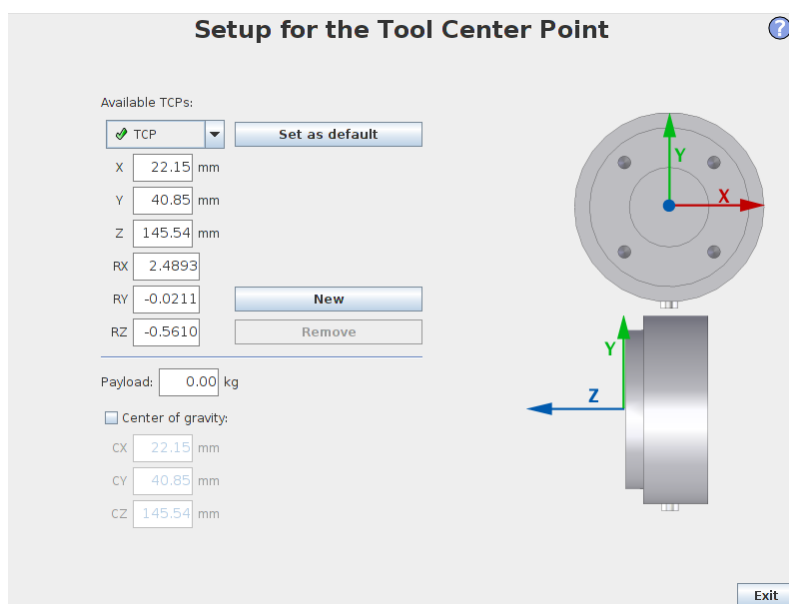


Slika 13: Spajanje tipkala na kontroler robota



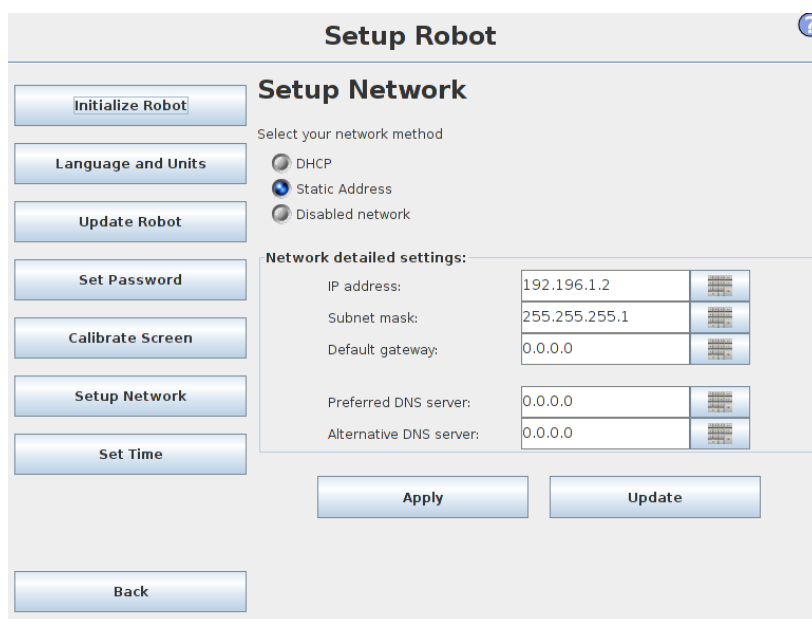
Slika 14: Kontroler robota s izvučenim tipkalima za precizno snimanje

Nešto ranije je spomenuto da UR3 nosi kameru, a s obzirom na to da bi za kvalitetu slike bilo dobro da se ostvari paralelnost sa podlogom, potrebno je što preciznije odrediti središnju točku alata (TCP). Za to je bila potrebna improvizirana naprava koja je simulirala udaljenost kamere od podloge (koje je konstrukcijski ograničena na cca. 15cm), a postupak određivanja TCP-a je taj da se šiljasti predmet „napada“ s tri strane robotom koji nosi alat. Nakon toga robot sam izračuna udaljenosti i vrijednosti koordinata, slika 15.



Slika 15: Središnja točka alata (kamere)

Budući da će UR3 komunicirati s drugim robotom (UR5), potrebno mu je namjestiti mrežne postavke. Njima se može pristupiti odmah iz početnog izbornika odlaskom na rubriku „network setup“ u kojoj se odabire statička adresa. Također, potrebno je paziti kako se ista adresa ne bi dodjelila na oba robota.



Slika 16: Mrežne postavke UR3 robota

2.4.2 Universal robots UR5

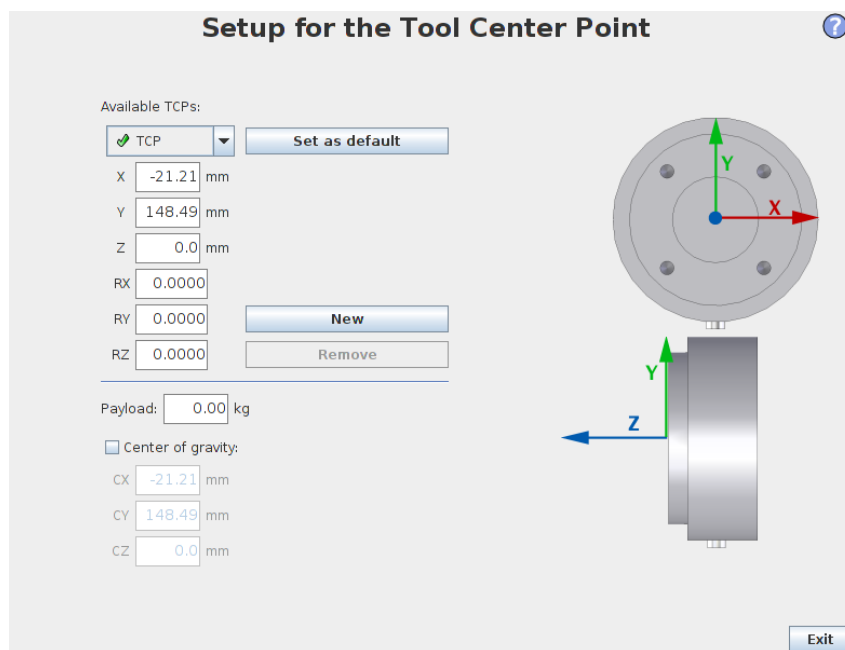
Baš kao i njegova slabija izvedba UR3, UR5 je industrijski robot široke primjene nosivosti do 5kg i dosega 850 mm. Posjeduje i sve certifikate kao i njegova manja izvedba, također, ima 16 digitalnih ulaza i izlaza, 2 analogna ulaza i izlaza. U ovom sustavu UR5 je zamišljen kao manipulator svjetla i rukovodeći član procesa tj. on upravlja ostalim elementima sustava. Shodno tome, optički senzori (jedan za kraj pokretne trake te jedan za početak), kao i tipkala za fino podešavanje slike spojeni su na digitalne ulaze, s druge strane upravljački signali pokretne trake spojeni su na digitalne izlaze dok je jedan analogni ulaz postavljen kako bi kamera preko 12-pinskog kabela koji je spojen na enkoder mogla programski¹ dati signal kada je snimanje završeno kako bi se traka mogla zaustavljati i promijeniti smijer.



Slika 17: Robot UR5 s improviziranom hvataljkom za svjetlo

¹ Korišten je unaprijed razvijeni C++ kod za obradu snimljenih fotografija

Jednako kao i kod UR3, nužno je dobro odrediti okomitost i paralelnost s obzirom na predmet snimanja, u ovom slučaju lim. Da bi se to ostvarilo, libelom se postiže ravnost i paralelnost s pokretnom trakom. Kako UR5 preko imporviziranog nosača nosi svjetlo, potrebno je računski pomoću trigonometrije odrediti središnju točku alata. Iz slike 18. se da primjetiti kako nije postavljen pomak u z osi, što zapravo nije ni bitno, jer oko te osi robot rotira svjetlo, a to je ujedino i njegova jedina kretnja.



Slika 18: Središnja točka alata UR5

Kao što je već rečeno, da bi roboti mogli međusobno komunicirati potrebno im je postaviti mrežne postavke kod UR5 je postupak postavljanja jednak kao i kod UR3 pa ga nije potrebno iznova navoditi. Jedina razlika na koju treba paziti je da se IP adrese postave na istoj mreži te da im se ne dodjeli ista IP adresa, slika 19.

Setup Robot ?

Setup Network

Select your network method

☐ DHCP

☒ Static Address

☐ Disabled network

Network detailed settings:

IP address:	192.168.1.1	
Subnet mask:	255.255.255.1	
Default gateway:	0.0.0.0	
Preferred DNS server:	0.0.0.0	
Alternative DNS server:	0.0.0.0	

Apply **Update**

Back

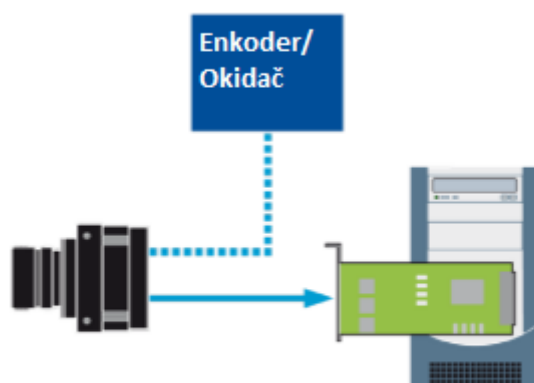
Slika 19: Mrežne postavke robota UR5

3. TEHNIČKE SPECIFIKACIJE SUSTAVA

Ako se bolje prouči slika 1, vidljivo je da okosnicu sustava čini robot UR5. Njegova zadaća je upravljanje pokretnom trakom kao i komunikacija sa senzorima te robotom UR3. Kako je limeni profil izuzetno tanak, on sam nije u mogućnosti aktivirati optičke senzore stoga se priskočilo uvođenju još nekih pomagala kao što su komadi aluminijskih profila kako bi aktivirati optičke senzore.

3.1 POSTAVKE SLIKE I KAMERE

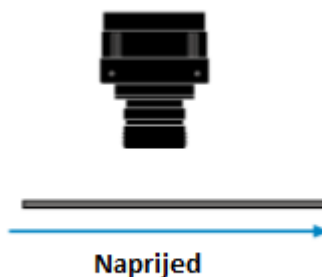
Kako se uzima više slika istog područja samo pod različitim kutovima, potrebno je transportni sustav (konvejer) pokretati u ova smjera. Ako se tome pribroji da je enkoder montiran na motor koji pokače konvejer, a on šalje okidajuće signale kameri dolazi se do zaključka kako će kamera, koja je iznad konvejera i lima koji snima, uzimati sliku u oba smjera kretanja konvejera.



Slika 20: Shema toka signala(isprekidana) i informacija(puna)

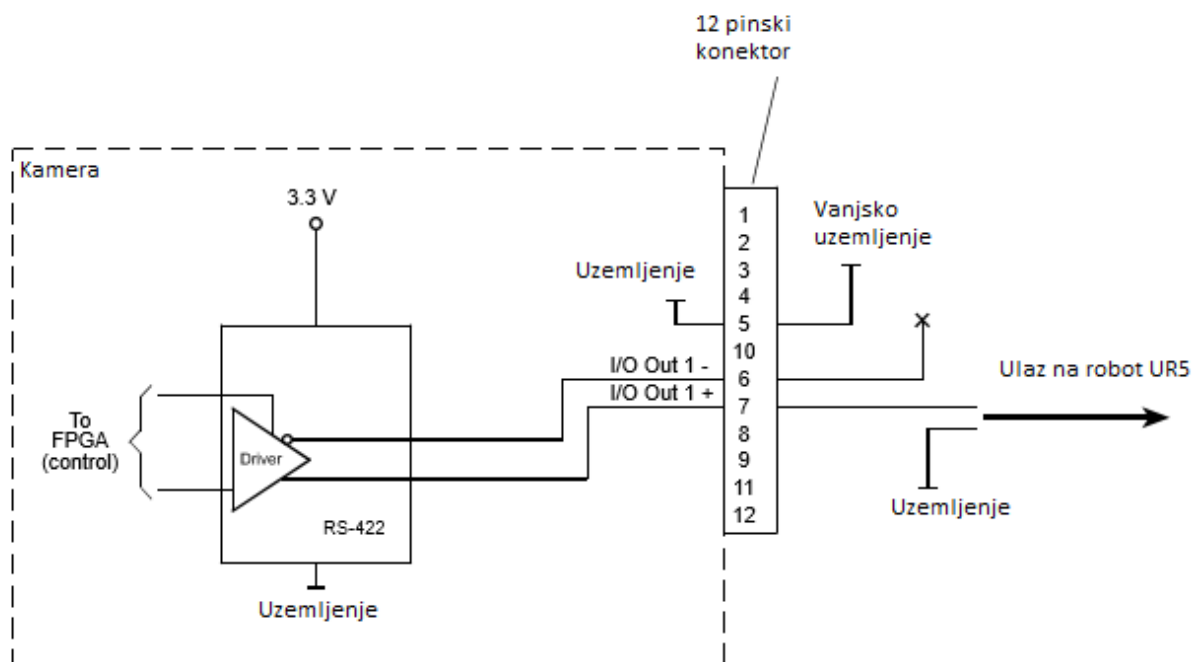
Takvo što je nepoželjno za naše rješenje pa se iz pogleda u literaturu [7] dolazi do spoznaje kako je moguće birati između snimanja samo u jednom smjeru (naprijed ili natrag) ili

snimanje u oba smjera. Za ovaj tehnički sustav nabolje rješenje je kretanje konvejera prema „gore“, odnosno prema naprijed.



Slika 21: Smjer snimanja slike

Postavke smjera okidanja, kao i druge postavke, npr. veličinu slike, otvor blende, posvjetljenje, itd. moguće je postaviti u programu Pylon kao i programsku u programskom jeziku C++ kao što je to napravljeno u ovom radu. Nakon postavljanja značajki okidanja slike potrebno je postaviti značajka i njenog zaustavljanja. Da bi se postigla ujednačenost u veličini snimljene fotografije, najbolji način je iskoristiti jedan analogni izlaz koji ima kamera, a koji je preko 12-pinskog kabela izvučen kod enkodera i spojen kao analogni ulaz na robota UR5.

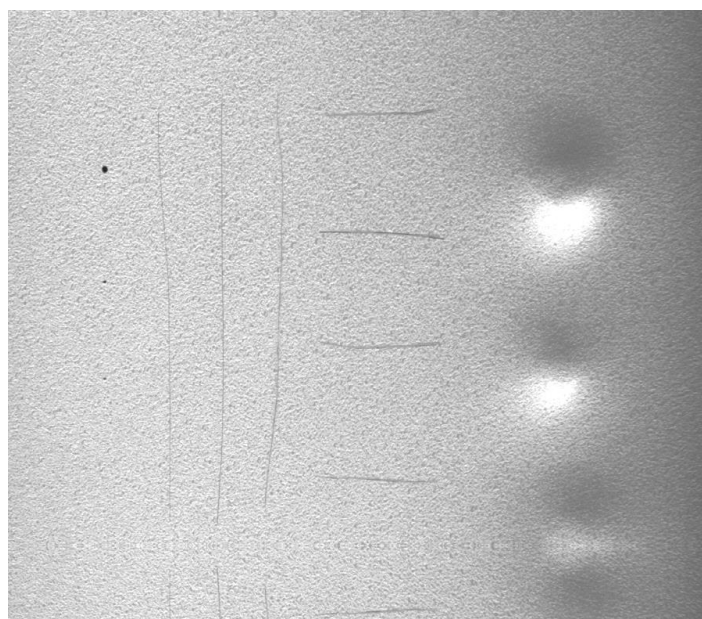


Slika 22: Signali s kamere

Testiranjem je utvrđeno kako analogni signal na robotu u aktivnom stanju daje 3.3 V pa je stoga za program kao granica okidanja bila vrijednost od 2.7 V. Kako bi kamera poslala analogni signal potrebno je programski napisati komandu koja će nakon uzimanja slike aktivirati signal:

```
Camera.AcquisitionStart.Execute();  
Camera.LineSelector.SetValue(LineSelector_Out2);  
Camera.LineInverter.SetValue(true);
```

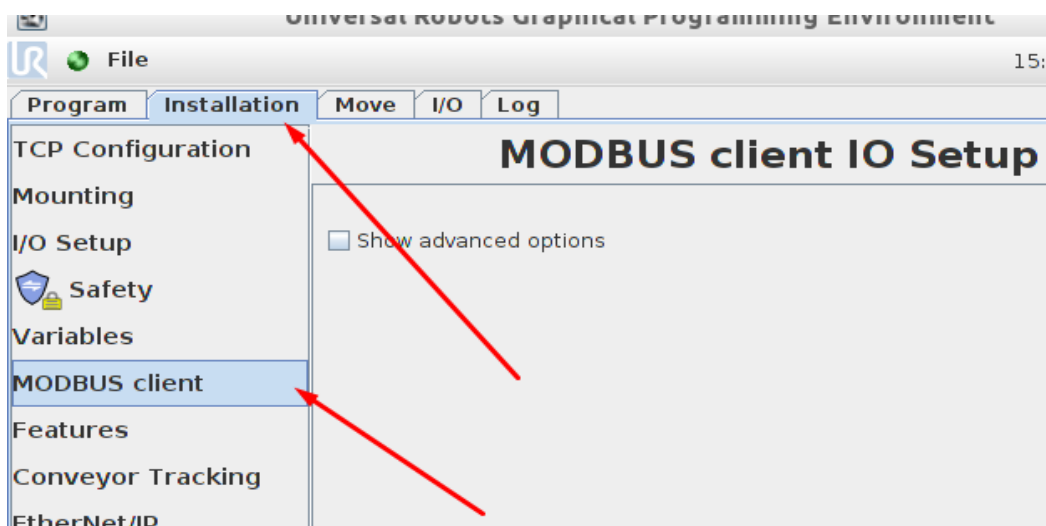
Kao sredstvo sigurnosi iza predmeta snimanja je postavljen profil kako bi se u slučaju rušenja programa konvejer ipak zaustavio. Nakon što su se sve značajke kamere definirane, može se reći kako se iz jednog pukog elementa za snimanje dobila primjenjiva komponenta koja udovoljava svim potrebama ovog sustava.



Slika 23: Primjer slike nakon postavljanja parametra

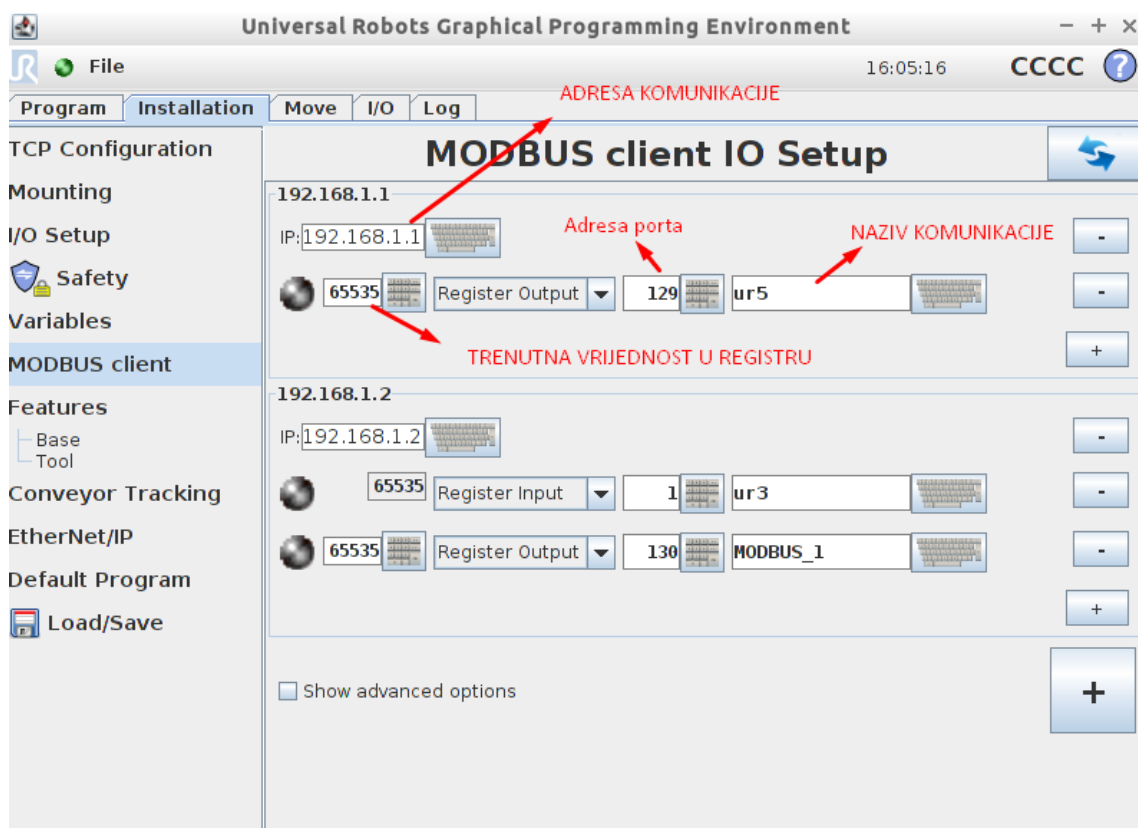
3.2 KOMUNIKACIJA IZMEĐU ROBOTA

S obzirom na to da roboti rade naizmjenični posao manipuliranja elementima njihova komunikacija je okosnica programa. Ona se postiže postavljanjem Modbus komunikacije. Kao što je prikazano na slici 1. kontroleri robota su međusobno spojeni ethernet kabelom što je preduvjet za postavljanje Modbusa. Nakon što se spoje kontroleri i postave mrežne postavke robota na način koji je predstavljeno u poglavlju o robotima može se preći na postavljanje komunikacije. U izbornoj traci pod „Installation“ nalazi se rubrika MODBUS client, odabirom na nju s desne strane otvara se prazan izbornik izbornik, slika 24.



Slika 24: Modbus klijent

U praznom izborniku s desne strane nalazi se veliki znak „+“. Odabirom na njega otvara nam se prozor za unos adrese modbus komunikacije, ovdje je potrebno obratiti pozornost je adresa komunikacije mora biti na istim adresama kao i adrese robota.



Slika 25: Postavljanje modbus komunikacije

Sljedeća stavka na koju je potrebno obratiti pozornost su adrese ili broj porta. Kako literatura [8] i [9] za komunikaciju preporuča rad s registrima, na raspolaganju je široka paleta 16-bitnih registrata kojima je unaprijed određena namjena.

Address	Pre 3.0	3.0	3.1	R=Read, W=Write		
				R	W	
0	x	x	x	*		Inputs, bits 0-15 [BBBBBBBBTTxxxx] x=undef, T=tool, B=box
1	x	x	x	*	*	Outputs, bits 0-15 [BBBBBBBBTTxxxx] x=undef, T=tool, B=box
2	x	x	x		*	SetOutputsBitsMask 0-15 [BBBBBBBBTTxxxx] x=undef, T=tool, B=box
3	x	x	x		*	ClearOutputsBitsMask 0-15 [BBBBBBBBTTxxxx] x=undef, T=tool, B=box
4	x	x	x	*		Analog input 0 (0-65535)
5	x	x	x	*	*	Analog input 0 domain e {0=current[mA], 1=voltage[mV]}
6	x	x	x	*		Analog input 1 (0-65535)
7	x	x	x	*	*	Analog input 1 domain e {0=current[mA], 1=voltage[mV]}
8	x	x	x	*		Analog input 2 (tool) (0-65535)
9	x	x	x	*	*	Analog input 2 (tool) domain e {0=current[mA], 1=voltage[mV]}
10	x	x	x	*		Analog input 3 (tool) (0-65535)
11	x	x	x	*	*	Analog input 3 (tool) range e {0=current[mA], 1=voltage[mV]}
16	x	x	x	*	*	Analog output 0 output (0-65535)

Slika 26: Adrese portova i njihova namjena

Kako je komunikacija zamišljena na način da roboti međusobno izmjenjuju numeričke podatke kao sredstvo oznake stanja u kojem se nalaze oba robota imaju jedan registar za ulaz (čitanje) podataka i jedan izlaz (zapis) podatka. UR5 također ima još jedan izlazni registar prema UR3, no to će se detaljnije razraditi kasnije. Iz slika 26 i 27 da se zaključiti kako nisu svi registri dostupni za pisanje i čitanje podataka, stoga su za komunikaciju uzeti registri na adresama 1, 129 i 130.

34-127	x	x	x			Reserved for future system variables				
128-255	x	x	x	*	*	General purpose 16 bit registers				
256-	x	x	x	*		Robot state				
512-	x	x	x	*		Program state (not implemented)				
768-	x	x	x	*		Tool states				
1024-	x	x	x	*		GUI state (not implemented)				
2048-	x	x	-	*		RT Machine control				

Slika 27: Odabrane adrese registara

Komunikacija između robota detaljnije će biti objašnjena u samom programu robota (pogledati prilog), no za ovo poglavlje bitno je izdvojiti stanja koja su robotima ključna za komunikaciju.

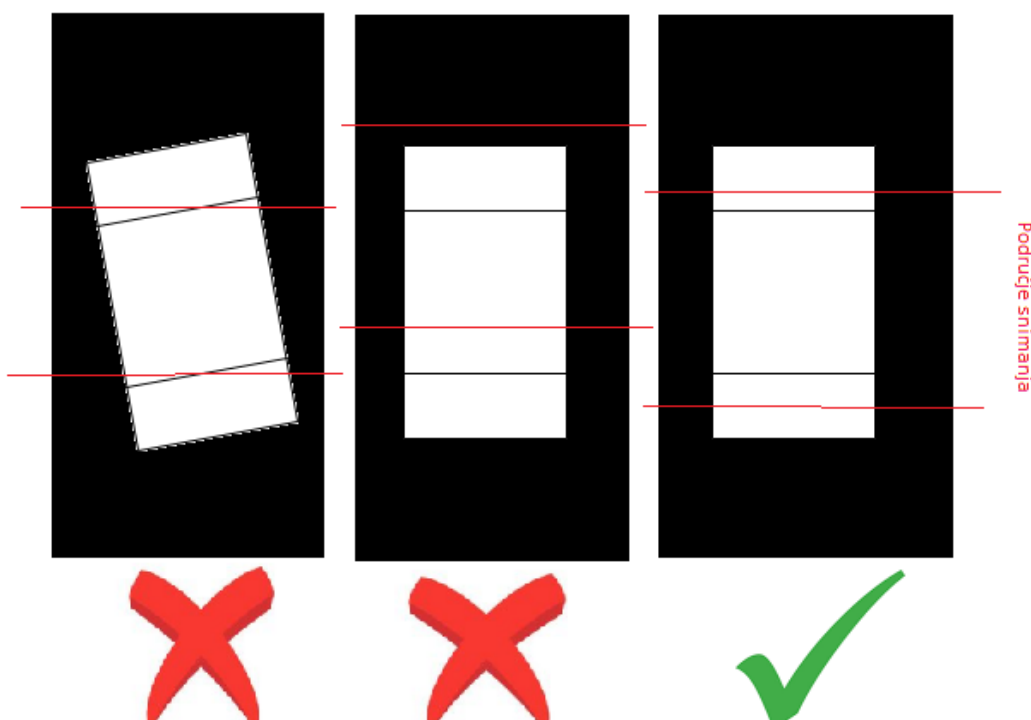
Tablica 2 : Prikaz komunikacije između robota

Universal_robot UR5			Universal_robot UR3	
1	Set UR5= 9	Inicijalizacija robota UR5	Set UR3=9	Inicijalizacija robota UR3
2			Wait UR5≠1	Čekanje da se upali robot UR5
3	Wait UR3 ≠ 2	Provjera komunikacije između robota	Wait UR5 = 1	Čekanje signala da robot UR3 može zauzeti početni položaj
4	Set UR5=1	Znak robotu UR3 da robot UR3 ode u početni položaj		
5	Wait ur3=2	čekanje signala da je UR3 zauzeo potrebni položaj		
6			Set UR3= 2	Znak da je UR3 zauzeo potrebni položaj
7			Wait UR5 = 4	Čekanje kraja snimanja
8	Set UR5= 4	Znak da je snimanje završeno		
9	Wait UR3 = 8	Čekanje signala da je UR3 zauzeo poziciju idućeg kuta		
10			Set UR3=8	Znak da je UR3 promijenio kut

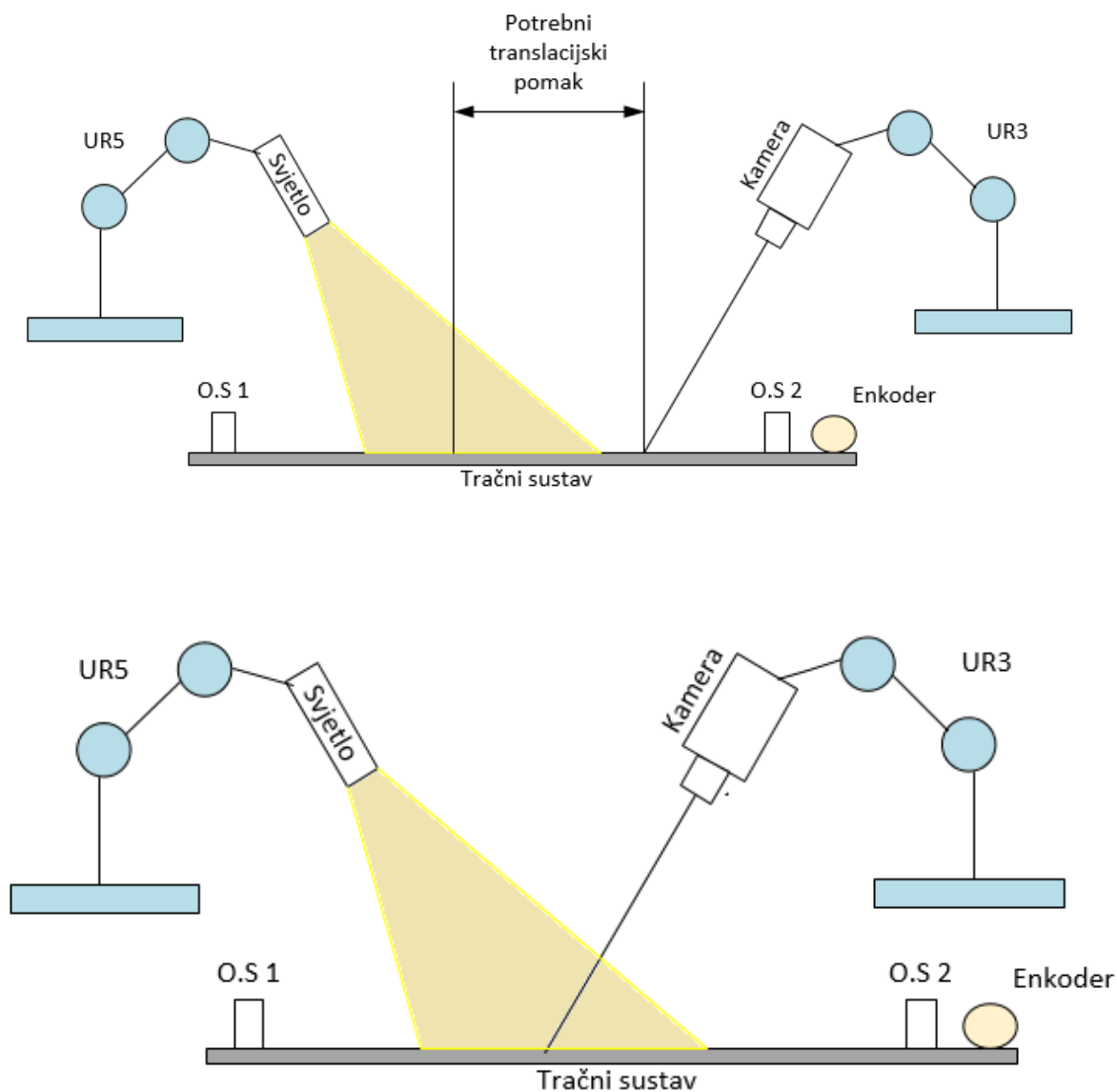
Pokretanje sustava je zamišljeno tako da se prvo uključi program na robotu UR3, a tek nakon njega UR5. Nakon početne inicijalizacije stanja u registrima i prvog prolaska kroz program proces se ponavlja od koraka 4 do koraka 10 sve dok se snimanje ne završi.

3.3 KOREKCIJA POMAKA USLJED ROTACIJE SVJETLA

Nešto ranije u ovom radu je spomenuto kako robot UR5 nije bio u mogućnosti sam nositi svjetlo pa je ono iz tog razloga bilo uležišteno na jednom kraju za konstrukciju u laboratoriju. S obzirom da je za kvalitetu snimljene slike bitno da svjetlo pada na isto područje potrebno je uzeti u obzir translacijski pomak „svjetla“ uslijed promjene njegova kuta. Uz to na testnim limovima su s dvije paralelne linije označena područja koja sadrže greške kako bi se kasnije programski slika mogla izrezati samo to područje. Program funkcionira tako da pronađe dvije paralelne linije na slici te samo područje između njih uzima u promatranje. Kako je podloga konvejera crna, u slučaju da ona dospije u područje snimanja, program će misliti da je to prva crna linija te uzeti krivo područje u razmatranja. Isto tako, ukoliko se ne ostvari paralelnost s kamerom, kamera neće pronaći paralelne linije. Ispravno područje je prikazano na slici 28.

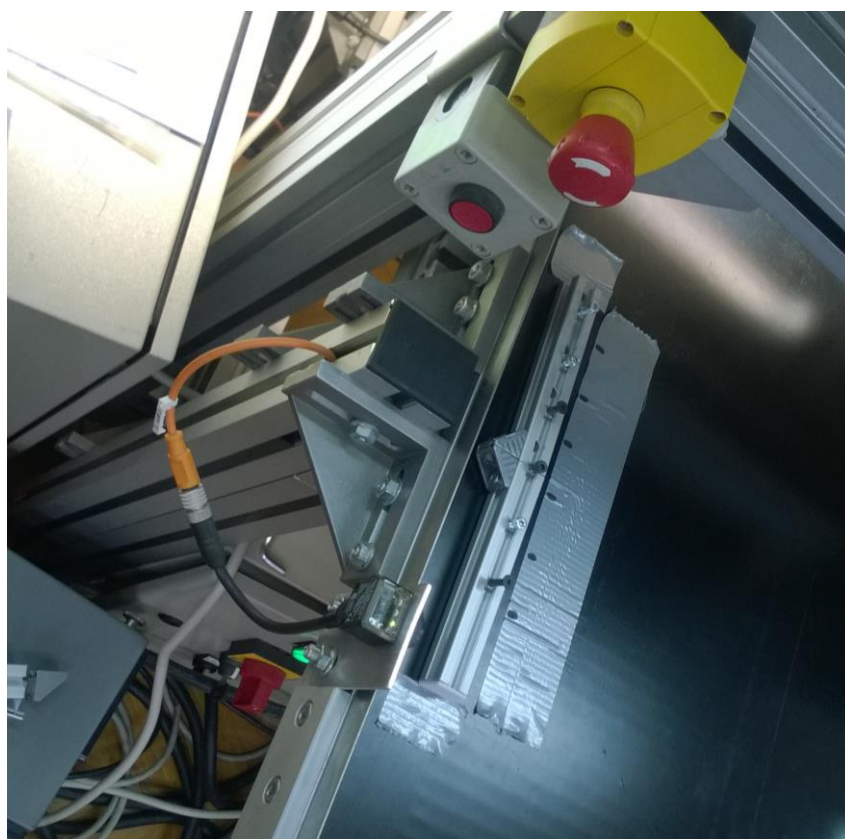


Slika 28: Pogreške kod snimanja zbog crne podloge



Slika 29: Neispravno i ispravno podešeno snimanje

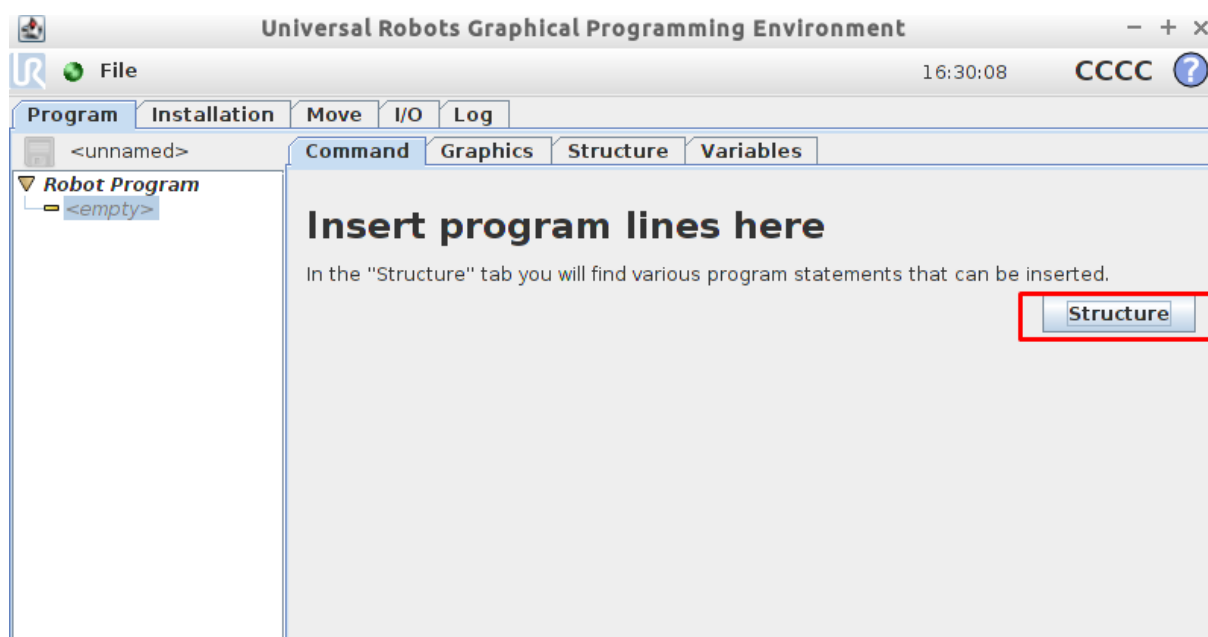
Da bi svijetlost uvijek padala na promatrano područje, robot UR3 mora kameru translirati za pomak koji je određen kutom rotacije svjetla. Trigonometrijskim proračun se dolazi do podatka od 35 mm. To je translacija koju robot kamere treba napraviti za svaki pomak kuta kamere od 10° kako bi održao pravilno područje snimanja. Iz tog razloga je napravljen jedan improvizirani aluminijski profil s vijcima postavljenim na razmacima od 35 mm, vijci služe kao zastavice te je u programu robota UR5 napisan dio programa koji stalno broji vijke koji prođu senzor te uz poznavanje kuta kamere određuje na kojem vijku odnosno zastavici će stati.



Slika 30: Letva s vijcima

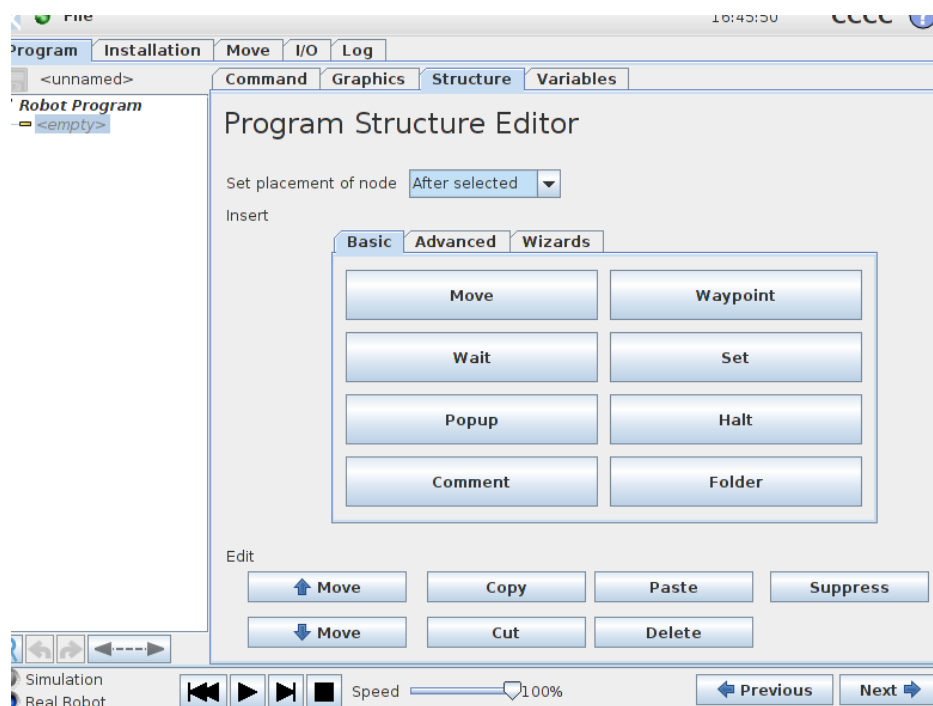
4. ALGORITAM I PROGRAMI

Nešto ranije je spomenuto kako će u ovom radu biti obrađena 4 programa na robotima, po jedan par programa za automatsko i precizno snimanje na oba robota. Algoritamski će se samo objasniti princip rada za automatsko snimanje. Programi su napisani pomoću grafičkoj upravljačkog sučelja na privjesku za učenje Universal robotsa (Polyscope).



Slika 31: Otvaranje liste programskih instrukcija

U ovom poglavlju prikazati će se neke od instrukcija korištene u programiranju te svi programi napisani instrukcijski poradi bolje preglednosti. Program se piše slaganjem naredbi iz više pripadajućih izbornika, izbornik se poziva odabirom na opciju *Structure*, slika 30.



Slika 32: Prikaz nekih instrukcija

Move – naredba koja određuje kretanje zglobova robota kroz određene točke (*Waypointe*), postoje 3 vrste ove naredbe, *moveL* (linearno kretanje zglobova), *moveJ* (koristenje interpolacije zglobova, rezultira krivudavom putanjom), *moveP* (linearno gibanje uz konstantnu brzinu i kružnu putanju).

Waypoint – instrukcija koja odgovara točki u prostoru kroz koju robot prolazi, ta točka može biti fiksna, relativna ili zapisana u obliku vektora varijable. U ovom radu većinom su korištene varijabilni waypointi kako bi se mogao jednostavno promjeniti kutovi komponenata zbrajanjem vektora korištenjem funkcije *pose_add*.

```
var=p[0.5,0,0,3.14,0,0.15]
```

Prve tri varijable označavaju kordinate *x*, *y*, *z*, dok su druge tri varijable njihove rotacije *rx*, *ry*, *rz*.

Wait – naredba kojom se označava čekanje određeno vrijeme ili pak neki signal s ulaza ili izlaza.

Wait

Please select what should trigger the robot's next action;

☐ No Wait

☐ Wait seconds

☐ Wait for Digital Input

☐ Wait for mA

☐ Wait for

Slika 33: Opcije naredbe Wait

Set – pokriva skupinu naredbi kojom se upravljaju digitalni izlazi ili se pridodaje vrijednost nekom izlazu npr. Modbus registru. Također uključuje i opciju *no action*, kao i opciju i za postavljanje nosivosti robotske ruke.

Set

Select the action you wish the robot to perform at this point in the program. You can also specify changes in the robot's payload.

☐ No Action

☐ Set Digital Output

☐ Set Analog Output mA

☐ Set

☐ Increment installation variable by one:

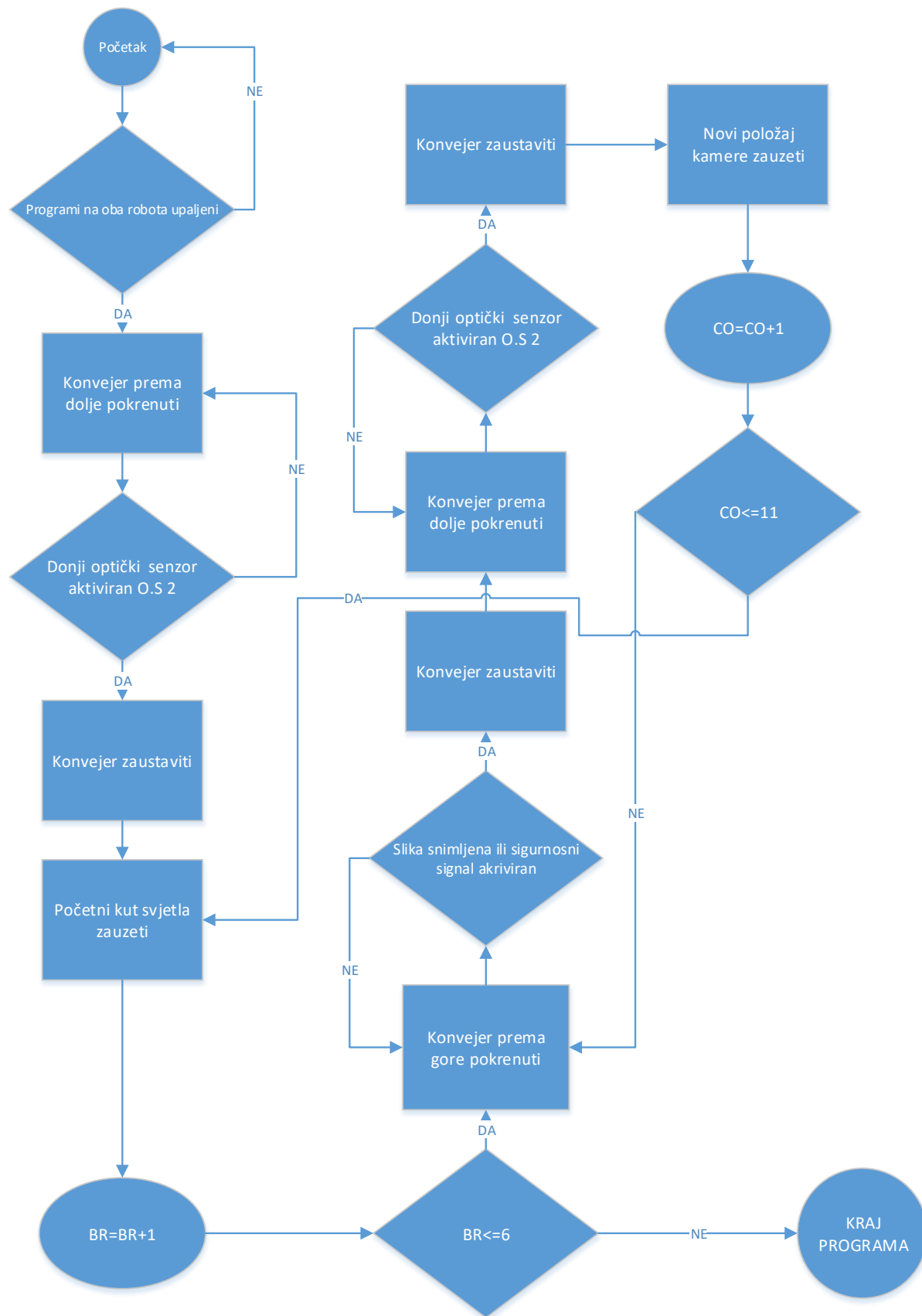
☐ Set the total payload to kg

☐ Set TCP

Perform action now

Slika 34: Opcije naredbe Set

4.1 ALGORITAM AUTOMATSKOG PROGRAMA



4.2 PROGRAM ZA AUTOMATSKO SNIMANJE NA ROBOTU UR5

```

var_5=0 // Varijabla brojila zastavice na donjem senzoru
var_6=1
var_1=p[0,0,0,0,0,0]
var_2=p[0,0,0,0,0,0.175] // Vektor kutnog pomaka svijetla (rotiacija
oko z osi)
var_3=ur3 // pridodavanje Modbus registra varijabli
var_4=ur5
Set ur5=9 // Inicijalizacija modbus komunikacije
Wait ur3 != 2 // Provjera komunikacije s robotom UR3
Set DO[3]= On // Pokretanje transportne trake u poziciju za početak
snimanja
Wait var_5=6 // Čekanje da predmet snimanja dođe na točnu poziciju
Set DO[3]=OFF // Gašenje transportne trake
var_5=0 // Resetiranje brojila za određivanje korekcije
pomaka na senzoru 0.S2
MoveL // Zauzimanje početnog kuta snimanja
    Waypoint_1
Loop 6 times (LOOP_1) // Petlja za promjenu kuta svjetla
    var_2=pose_add(var_2,var_1) // Zbrajanje vektora (određivanje
promjene kuta)
    MoveL // Promjena u novi kut snimanja
        var_2
    Loop 11 times (LOOP_3) // Petlja za promjenu kuta kamere
        var_5=0//Resetiranje brojila za određivanje korekcije
pomaka na senzoru 0.S2
        Set ur5=1 //Slanje znaka robotu UR3 da zauzme početni
položaj
        Wait ur3=2 // Čekanje signala da je UR3 u potrebnom
položaju
        Set DO[2]=ON // Pokretanje transportne trake u smjeru
snimanja
        Wait (analog_in[1]>2.7) or (digital_in[4]=true) // Čekanje
kraja snimanja ili aktiviranje sigurnosnog senzora
        Set DO[2]=OFF // Stopiranje transportne trake
        Set DO[3]=ON // Povratak transportne trake u početni
položaj
        if Loop_1=0 // DIO KODA KOJI SLUŽI ZA KOREKCIJU
POMAKA USLIJED PROMJENE KUTA SVJETLA
            if Loop_3<10
                Wait var_5=6
            else Wait var_5=5
        else if Loop_1=1
            if Loop_3<10
                Wait var_5 = 5

```

```
        else Wait var_5 = 4
    else if Loop_1 = 2
        if Loop_3<10
            Wait var_5 = 4
        else Wait var_5 = 3
    else if Loop_1 = 3
        if Loop_3<10
            Wait var_5 = 3
        else Wait var_5 = 2
    else if Loop_1 = 4
        if Loop_3<10
            Wait var_5 = 2
        else Wait var_5 = 1
    else if Loop_1 = 5
        if Loop_3<10
            Wait var_5 = 1
        else Wait var_5 = 0
    else if Loop_1 = 6
        if Loop_3<10
            Wait var_5 = 1
    Set DO[3]=OFF // Znak da je predmet snimanja na
ispravnoj poziciji za drugo snimanje
    Set ur5=4 // Slanje signala robotu UR3, da je snimanje
završeno te da zauzme novi kut kamere
    Wait ur3=8 // Čekanje signala da je robot UR3 zauzeo novi
kut snimanja
MoveL // Povratak u početni položaj
    Waypoint_1

HALT //Prekid programa

Thread_1 // Dio stalnog programa za korekciju pomaka usljed
promjene kuta svjetla
    Wait (digital_in[3]= True) and (digital_out[3]=True)
    Wait DI[3]=L0
    var_5=var_5+var_6
```


4.3 PROGRAM ZA AUTOMATSKO SNIMANJE NA ROBOTU UR3

```

var_7=p[0.035,0,0,0,0,0,0] //Varijable za korekciju pomaka
var_4=p[0.035,0,0,0,0,0,0]
var_6=p[0,0,0,0,-0.0875,0] // Varijabla za pomak kuta kamere od 5*
var_5=p[0,0,0,0,0,0]
var_2=ur5 // Pridodavanje Modbus registra varijabli
var_3=ur3
Set ur3=9 //Inicijalizacija robota
Wait ur5 != 1 // Čekanje pokretanja robota UR5
Wait ur5 =1 // Čekanje signala da robot UR3 može zauzeti početni
položaj
Loop 6 times (LOOP_2) //Petlja za praćenje pomaka kuta svjetla
    MoveL // Postavljanje u početni položaj
        Waypoint_1
        var_6=p[0,0,0,0,-0.0875,0]
    if LOOP_2=1
        var_8=pose_add(var_4, var_7)
        MoveL
    if LOOP_2>1
        var_8
        var_8=pose_add(var_8,var_7)
        MoveL
        var_8
    Loop 11 times // Petlja za promjenu kuta kamere
        Wait ur5=1 // Čekanje signala da robot UR3 zauzme novi kut
        Set ur3=2 // Znak da je UR3 zauzeo početni položaj
        Wait ur5=4 // Čekanje kraja snimanja
        var_6=posea_add(var_5,var_6) // Računanje novog kuta
kamere
        MoveL // Pomicanje u novi kut kamere
        var_6
        Set ur3=8 //Slanje signala da je novi kut kamere
postavljjen
        Wait 1.0s
MoveL // Vraćanje u početni položaj
    Waypoint_1
HALT // Zaustavljanje programa

```

Dio koda za korekciju
pomaka uslijed promjene
kuta svjetla

4.3 UR5 PROGRAM ZA PRECIZNIJE POZICIONIRANJE

```

var_7=0
var_8=1
var_5=0
var_6=1
var_1=p[0,0,0,0,0,0]
var_2=p[0,0,0,0,0,0.175]
var_3=ur31
var_4=ur5
Set ur5=9
Wait ur3!=2
Set D0[3]=On
Wait var_5=6
Set D0[3]=Off
var_5=0
MoveL
    Waypoint_1
Loop 6 times (LOOP_1)
    var_2=pose_add(var_2,var_1)
    MoveL
        var_2
    var_7=var_7+var_8 //Brojač promjena kuta svjetla
    Set MODBUS_3 = var_7
    Wait 2.0s
    if digital_in[5]=True // Ukoliko je sklopka za zaustavljanje
kuta snimanja uključena
        Loop 11 times (LOOP_4)
            if digital_in[4]=True // Ukoliko je sklopka za
zaustavljanje kuta snimanja uključena
                Call SubProgram_1
                var_5 = 0
                Set D0[2] = Off
                Set D0[3] = On
                if Loop_1 = 0 // DIO KODA KOJI SLUŽI ZA
KOREKCIJU POMAKA USLIJED PROMJENE KUTA SVJETLA
                    if Loop_4<10
                        Wait var_5 = 6
                    else Wait var_5 = 5
                else if Loop_1 = 1
                    if Loop_4<10
                        Wait var_5 = 5
                    else Wait var_5 = 4
                else if Loop_1 = 2
                    if Loop_4<10
                        Wait var_5 = 4

```

//Inicijalizacija programa

```

        else Wait var_5 = 3
    else if Loop_1 = 3
        if Loop_4<10
            Wait var_5 = 3
        else Wait var_5 = 2
    else if Loop_1 = 4
        if Loop_4<10
            Wait var_5 = 2
        else Wait var_5 = 1
    else if Loop_1 = 5
        if Loop_4<10
            Wait var_5 = 1
        else Wait var_5 = 1
    else if Loop_1 = 6
        Wait var_5 = 0
        if digital_in[4] = False // Ukoliko je
isključen sklopka za stalno kretanje trake
            Set D0[3] = Off
            Set ur5 = 4
        Else
            Set D0[3] = off
    Else // Ukoliko sklopka za zaustavljanje kuta nije uključena
nastavi mjenjati kut kamere
        NoAction
Thread_1 // Dio stalnog programa za korekciju pomaka usljed
promjene kuta svjetla
    Wait(digital_in[3] = True) and (digital_out[3] = True)
    Wait DI[3] = L0
    var_5 = var_5 + var_6

SubProgram_1
While digital_in[5]=Ture
var_5=0
Set D0[2]=Off
Set D0[3]=On
if Loop_1 = 0 // DIO KODA KOJI SLUŽI ZA KOREKCIJU POMAKA
USLIJED PROMJENE KUTA SVJETLA
    if Loop_4<10
        Wait var_5 = 6
    else Wait var_5 = 5
else if Loop_1 = 1
    if Loop_4<10
        Wait var_5 = 5
    else Wait var_5 = 4
else if Loop_1 = 2
    if Loop_4<10
        Wait var_5 = 4
    else Wait var_5 = 3
else if Loop_1 = 3

```

```
    if Loop_4<10
    Wait var_5 = 3
    else Wait var_5 = 2
else if Loop_1 = 4
    if Loop_4<10
    Wait var_5 = 2
    else Wait var_5 = 1
else if Loop_1 = 5
    if Loop_4<10
    Wait var_5 = 1
    else Wait var_5 = 1
else if Loop_1 = 6
    Wait var_5 = 0
if digital_in[5]=False
    Set D0[3]=Off
    Set ur5=4
Else
    Set D0[3]=off
```

4.4 UR3 PROGRAM ZA PRECIZNIJE SNIMANJE

```

kut=0
var_15=5
pp=0.0875
var_7=p[0,0,0,0,0.0875,0]
var_11=MODBUS_3
var_10=p[0,0,0,0,0.0875,0]
var_9=p[0,0,0,0,-0.0875,0]
var_4=p[0.35,0,0,0,0,0]
var_6 = p[0 0,0,0,-0.0875,0]
var_5=p[0,0,0,0,0,0]
var_2=ur5
var_3=ur3
Set ur3=9
Wait 2s
//promjeni kut
MoveL
    Waypoint_1
If var_11>1
    var_12=p[0.035,0,0,0,0,0]
    var_13=0.035
    var_8=var_11-1
    zz=var_8*var_13
    var_14=p[zz,0,0,0,0,0]
    MoveL
        var_14
    Wait: 5s
    Loop
        Wait(digital_in(4)=False ) or (digital_in(5)=False)
        If kut<=(45) and(digital_in[5]=false) // Ukoliko robot
dođe do krajnjih položaja vrati se u početnu poziciju zbog
sigurnosti
            MoveL
                Waypoint_1
                kut=0
            Else kut>=(5) and(digital_in[5]=False)
                MoveL
                    Waypoint_1
                    kut=0
                if digital_in[4]=False
                    MoveL
                        var_6
                if digital_in[5]=False
                    MoveL
                        var_7

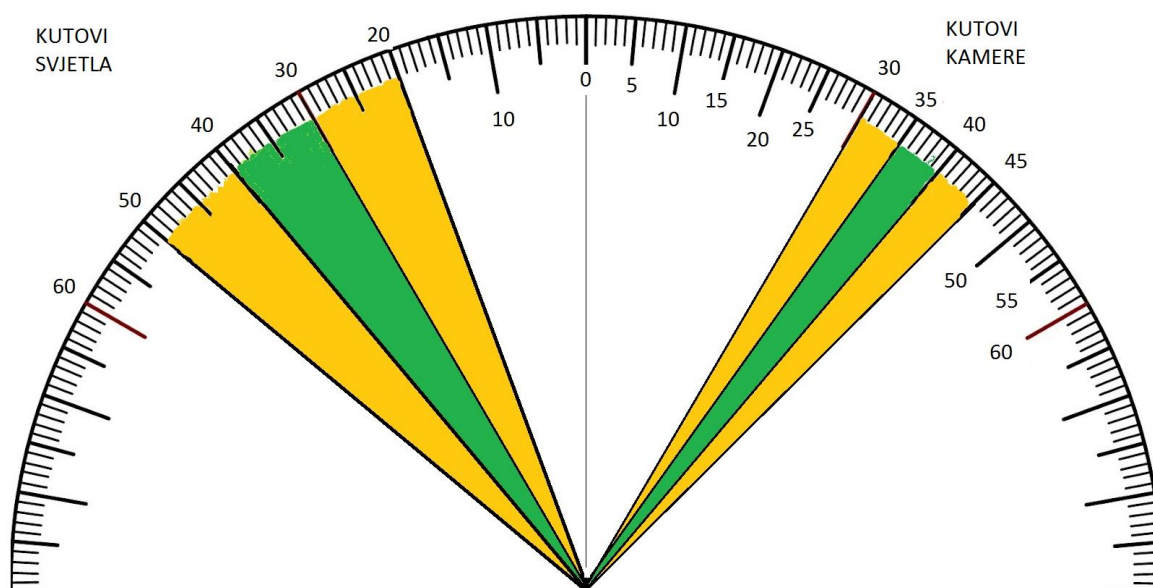
```

```
THREAD_1 // Stalni program za povećavanje kuta  
Wait DI[5]=LO  
Wait DI[5]=HI  
kut=kut+var_15
```

```
THEAD_2 // Stalni program za smanjivanje kuta  
Wait DI[4]=LO  
Wait DI[4]=HI  
kut=kut-var_15
```

5. REZULTATI SNIMANJA

Nakon odrađenog programskog dijela, potrebno je napraviti testno snimanje. U prvom pokretanju programa za automatsku inspekciju zabilježe se kutovi koji prikazuju najbolju sliku. Najbolja slika je ona na kojoj se greške najbolje vide. Kada se sazna pod kojim kutovima se nalaze najbolje slike, programima za precizno snimanje odredimo korak kuta koji služi za finije pokame kamere u razmacima od 1° do 5° kako bi precizno odredili kut pod kojim kamera snima komad lima. Na slici 35 su označena područja s najboljim rezultatima snimanja.



Slika 35: Kutovi s najboljim rezultatima

Sama vizijska inspekcija snimljene fotografije radi se na način da kroz sliku „protrči“ jedna matrica koja zagladi sve površine iste boje (područje bez grešaka), dok ekstremi odnosno greške budu jasnije vidljivi. Unaprijed pripremljeni C++ program za vizijsku inspekciju, snima svaku uhvaćenu fotografiju pa se tako uvidiom u slike može vidjeti napredak i poboljšanja. Slike 36 do 39 prikazuju poboljšane snimljene fotografije i rezultate na testnom program za vizijsku inspekciju.

**Slika 36: Primjer snimanja 1**

Slika 36. Prikazuje nam loš kut snimanja, a program za detekciju grešaka teško detektira oštećenja.

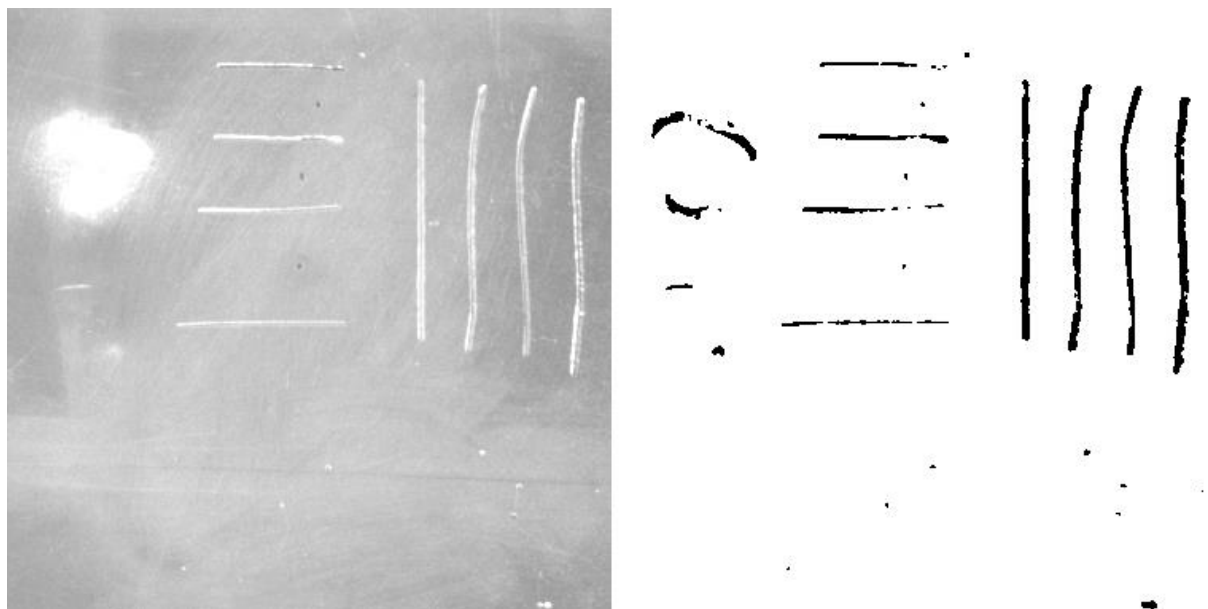
**Slika 37: Primjer snimanja 2**

U odnosu na sliku 36 da se primjetiti da su ovdje oštećenja puno vidljivija, no u desnom rubu je još uvijek velika količina svjetlosti koja prikriva greške.



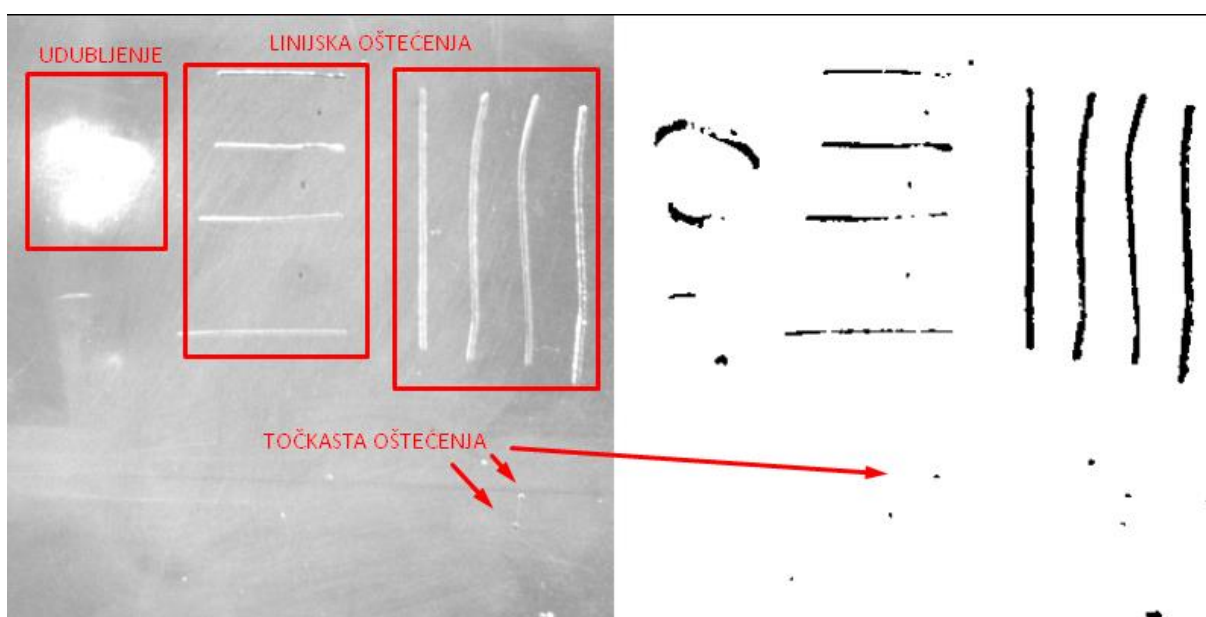
Slika 38: Primjer snimanja 3

Preciznijom promjenom kuta otkriva se desni dio slike i primjećuje se jedno oštećenje koje do sada nije bilo vidljivo, međutim odsjaj je na desnom rubu još uvijek prejak da bi program za detekciju oštećenja to mogao kasificirati kao grešku.



Slika 39: Finalna fotografija

Finalna fotografija daje detaljan uvid u sve greške, ovisno o vrsti primjene ove inspekcije moguće je napraviti tolerancijsko područje za svaku grešku. Ukoliko se napravi analiza fotografije, primjećuje se više vrsta oštećenja. Na slici 40. vidljive su 3 skupine oštećenja: udubljenje, linijska te točkasta oštećenja. Kroz proces ispitivanja primjećeno je kako su udubljenja izraženija pod većim kutovima pa je tako za sustav kojem je najveći problem ta skupina oštećenja bolje koristiti veće kuteve snimanja, no kako se ovdje radi o sveobuhvatnom snimanju tražena su rješenja koja najbolje prikazuju sva oštećenja.



Slika 40: Analiza snimljene fotografije

6. ZAKLJUČAK

Kako u svijetu postoji rastući trend estetike kao najbitnijeg prioriteta kod kupovine nekog proizvoda, veliku zastupljenost u izradi takvih proizvoda imaju i razne vrste limova. Ukoliko lim s greškom uđe u proizvodnju to najčešće rezultira s neispravnim proizvodom, a kako se takve greške teško uočavaju velika je vjerojatnost da će takav proizvod završiti na tržištu te na kraju rezultirati reklamacijom što stvara dodatne probleme proizvođaču. Visoki napredak u tehnologiji posljednjih godina ponudio nam je mogućnost vizijske inspekcije limenih bala prije početka proizvodnje.

Upravo iz toga proizlazi potreba za ovim radom. Iako vizijska inspekcija suštinski nije tema ovoga rada, određivanje njenih parametara je. U njemu su sadržane sve značajke na koje treba obratiti pozornost kako bi se dobili što bolji rezultati. Ovaj tehnički sustav uvelike olakšava potragu za optimalnim parametrima jer je velikim dijelom automatiziran, što smanjuje vrijeme potrebno za razvoj sustava vizijske inspekcije kod određenog problema. U radu su prikazani i neki od problema, no pravilnom optimizacijom tehničkog sustava uspjeli su se otkloniti. Na slici 36 zeleno su označena područja kuteva koji su za ovu primjenu pokazali kao najbolji.

Kako su dobiveni rezultati u kombinaciji s unaprijed razvijenim kodom za vizijsku inspekciju u konačnici dali zadovoljavajuće rezultate, može se reći kako su svi zadani zahtjevi korektno ispunjeni. Veliko zadovoljstvo pruža činjenica da se postojeća konfiguracija uz male preinake može koristiti za potrage optimalnih parametara i kod drugih industrijskih zahtjeva za vizijsku inspekciju.

LITERATURA

- [1] Racer GigE Handbuch user manual
- [2] Vision systems, [Choosing led lighting and controllers for machine vision systems](#)
- [3] [Lighting techniques for machine vision](#)
- [4] Enkoder Datasheet RI58-Oen
- [5] IRF, <https://ifr.org>
- [6] Universal robots UR3 & UR5 manual
- [7] [How to Set Up Several Basler racer Cameras in One Inspection Row](#)
- [8] [Universal robots modbus communication channels](#)
- [9] [Universal robots modbus ports](#)

PRILOZI

- I. CD-R disc
- II. Tehnička dokumentacija